CONDITIONS OF RELEASE

0061791 BR-112706

************************* U

************************* Y

Reports quoted are not necessarily available to members of the public or to commercial organisations.

Royal Signals and Radar Establishment
Memorandum 4342

On Networks, Optimised Feature Extraction and the Bayes Decision.

David Lowe and Andrew R. Webb

5th December 1989.

Abstract

In this paper we address the problem of multi-class pattern classification using adaptive layered networks. We view such networks as performing generalised linear discriminant analysis in which a particular parametric form is assumed for the nonlinear functions. Training the network consists of a least-square approach which combines a generalised inverse computation to solve for the final layer weights, together with a nonlinear optimisation scheme to solve for parameters of the nonlinearities. Such an approach performs feature extraction and classification simultaneously, in which the feature extraction is (optimally) matched to the classification scheme. We derive a general analytic form for the feature extraction criterion and interpret it for specific forms of target coding and error weighting. A particular aspect of the approach is to exhibit how *a priori* information regarding nonuniform class membership, uneven distribution between train and test sets and misclassification costs may be exploited in a regularised manner in the training phase of networks.

Contents

1 Introduction

Connectionist models based on adaptive layered networks (*e.g.* Multilayer Perceptrons [17] and Radial Basis Functions [2]) have been used with some success when operating as static pattern classifiers in problems as diverse as sonar classification [9], speech recognition [16] and medical diagnosis [1]. The ability of feed-forward layered networks to perform static pattern discrimination stems from their potential to create a *specific* nonlinear transformation into a space spanned by the outputs of the hidden units in which class separation is easier [21]. This transformation is constrained to maximise a specific feature selection criterion [21] which may be viewed as a nonlinear multi-dimensional generalisation of Fisher's Linear Discriminant function [5].

The expression for the network feature extraction criterion derived in [21] involves the *weighted* between class covariance matrix (where the weighting is determined by the *square* of the number of patterns in each class). This implies that adaptive networks trained on a 1-from-n classifier problem bias strongly in favour of those classes which have the largest membership in the training data. Thus, in order to minimise the error over the entire training set, the optimum solution for the network parameters is such that the network misclassifies patterns in classes with smallest representation in favour of those with larger representation in the training set, irrespective of the frequency of occurrence in actual 'operation'. Also, training of networks often takes no account of costs of misclassification.

These are undesirable features of networks (and many other standard classifiers) in problems where information on one particular class may be more difficult or expensive to obtain than other classes, and where it is important to consider the costs of misclassification. For instance, in speech recognition the bulk of the continuous acoustic signal consists of silence whereas the dominant information content is contained in the subword units ('phonemes'). Another example, which is considered in more depth in a separate paper [12] is in the problematic realm of medical prognosis: *given* a feature pattern as determined from a set of observations on a patient, what are the likely future health prospects of that patient. In this case the importance of misclassification could literally be a matter of life-and-death if resources had to be limited to those in most need who would gain maximum benefit.

These are illustrative of pattern classification problems where the distribution of patterns amongst the different classes in the training set is nonuniform and also follows a different distribution to the *expected* occurrence or relative importance of the classes in operation. In addition, there may be further prior knowledge which could be used to associate to each class a cost of misclassification with any other class.

Heuristic methods may be developed which attempt to compensate for some of these effects in training adaptive networks. For instance, the classes of the training data may be sampled according to a distribution which reflects the prior probability distribution and the network is subsequently trained on the sampled data. Alternatively, if training proceeds iteratively, the number of iterations in the learning cycle of a network may be varied for each class which would have a similar compensatory effect. Equivalently, the sum square error minimised by the network during training may be weighted by the frequency of occurrence of the patterns in each class.

In this paper, we adopt a least-squares approach to pattern classification using adaptive feed-forward classification networks. For a network with linear output units, we may view the final layer as performing a generalised linear discriminant function [4] with a specific parametric form for the nonlinearities. Optimisation of the network results in adaptation of the nonlinearities. For various combinations of target coding and error weighting schemes, we derive the feature extraction criteria which are being maximised by the network. The following section reviews the optimality of the least-squares approach and Section 3 places the adaptive feed-forward network within that framework. An expression for the network feature extraction criterion is derived and in Section 4 its properties are explored for various target coding and error weighting schemes.

2 Least–mean–squares Pattern Analysis

The least-mean-square-error design criterion has been widely used in pattern recognition since it assumes no prior knowledge of class distributions or *a priori* probabilities [3, 13, 22, 23]. In this section we shall review the properties of the least-mean-square approach which allow it to obtain an optimum set of weights for a generalised linear discriminant function in which the nonlinear functions are assumed known.

Consider a (possibly nonlinear) vector function $\boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_{n_0}(\boldsymbol{x}))^*$ of a pattern $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^*$. A generalised linear discriminant function , O, has the form [4, 10]

$$O = \lambda_0 + \sum_{j=1}^{n_0} \lambda_j g_j(\boldsymbol{x}), \tag{1}$$

where λ_j are the weights, λ_0 is a bias term and there are n_0 nonlinear functions, g_j, of the pattern $\boldsymbol{x}$. A pattern classification device such as this, employing a fixed nonlinear transformation $\boldsymbol{g}(\boldsymbol{x})$ followed by a linear transformation, has also been termed a *phi* machine [15, 24]. Several different forms for the nonlinear functions, $\boldsymbol{g}$, have been considered. For example, Specht [18] uses a polynomial discriminant function for pattern classification in which the coefficients are determined by an expansion of an approximation to the probability density function.

For a c–class problem, we may form c discriminant functions

$$O_k = \lambda_{0k} + \sum_{j=1}^{n_0} \lambda_{jk} g_j(\boldsymbol{x}), \qquad k = 1, \ldots, c \tag{2}$$

and for P patterns $\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^P$ we seek a solution for the parameters, $\{\lambda_{ij}, i = 0, \ldots, n_0, j = 1, \ldots, c\}$, which minimises the mean-square error

$$\begin{aligned} E &= \frac{1}{P} \sum_{i=1}^{P} \|\boldsymbol{\Lambda} \boldsymbol{g}(\boldsymbol{x}^i) + \boldsymbol{\lambda}_0 - \boldsymbol{t}^i\|^2 \\ &= \frac{1}{P} \|\boldsymbol{\Lambda} \boldsymbol{G} + \boldsymbol{\lambda}_0 - \boldsymbol{T}\|^2 \end{aligned} \tag{3}$$

where $\boldsymbol{G}$ is a $n_0 \times P$ matrix whose p-th column is $\boldsymbol{g}(\boldsymbol{x}^p)$; $\boldsymbol{T}$ is a $c \times P$ 'target' matrix with p-th column $\boldsymbol{t}^p = (t_1^p, t_2^p, \ldots, t_c^p)^*$, the target for the p-th pattern $\boldsymbol{x}^p$; $\boldsymbol{\Lambda}$ is the $c \times n_0$ matrix of weights ($\Lambda_{ji} = \lambda_{ij}$) and $\boldsymbol{\lambda}_0 = (\lambda_{01}, \lambda_{02} \ldots, \lambda_{0c})^*$ is the vector of biases. These functions $\boldsymbol{g}$ may be viewed as performing feature extraction according to some fixed rule and the parameters Λ and λ_0 are chosen in an optimal way. The target matrix is the matrix of desired outputs for the given set of training patterns.

There have been several different interpretations for the target matrix, T and these lead to different decision rules [22]. We shall consider the target matrix in more detail in Section 4.

Now, as the number of training samples, P, approaches infinity, and provided that the training samples are obtained at the same relative frequency as the occurrence of samples of the process, then n_i/P (where n_i is the number of samples in class i) tends to p_i, the prior probability and the limit of the error E is given by [3, 22]

$$E \rightarrow E_\infty = \sum_{i=1}^{c} p_i \langle \|\Lambda \boldsymbol{g}(\boldsymbol{x}) + \lambda_0 - \boldsymbol{s}_i\|^2 \rangle_i \qquad (4)$$

where the expectation is with respect to the conditional distribution of $\boldsymbol{x}$ on class i, i.e. for any function z of $\boldsymbol{x}$,

$$\langle z(\boldsymbol{x}) \rangle_i = \int z(\boldsymbol{x}) p(\boldsymbol{x}|i) d\boldsymbol{x}.$$

$\boldsymbol{s}_i = (s_{1i}, s_{2i}, \ldots, s_{ci})^*$ denotes the *prototype* target vector for class i (the columns of T are comprised of the vectors $\boldsymbol{s}_i$).

Thus, Equation (4) gives the large sample limit of the mean error. The solution for Λ and λ_0 which minimises E_∞ also minimises [3, 22]

$$E' = \langle \|\Lambda \boldsymbol{g}(\boldsymbol{x}) + \lambda_0 - \boldsymbol{\rho}(\boldsymbol{x})\|^2 \rangle, \qquad (5)$$

where the expectation is with respect to the unconditional distribution, $p(\boldsymbol{x})$, of $\boldsymbol{x}$ and $\boldsymbol{\rho}(\boldsymbol{x})$ is defined as

$$\begin{aligned} \boldsymbol{\rho}(\boldsymbol{x}) &= \sum_{i=1}^{c} \boldsymbol{s}_i p(i|\boldsymbol{x}) \\ &= \boldsymbol{S}\boldsymbol{p} \end{aligned} \qquad (6)$$

where $\boldsymbol{p}$ is the c–dimensional vector of posterior probabilities given the pattern $\boldsymbol{x}$ and $\boldsymbol{S}$ is the $c \times c$ matrix of prototype target vectors. Thus, $\boldsymbol{\rho}(\boldsymbol{x})$ may be viewed as a 'conditional target' vector : it is the expected target vector given a pattern $\boldsymbol{x}$, with the property that

$$\langle \boldsymbol{\rho}(\boldsymbol{x}) \rangle = \int_{-\infty}^{\infty} \boldsymbol{\rho}(\boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} = \sum_{i=1}^{c} p_i \boldsymbol{s}_i, \qquad (7)$$

the mean target vector. From Equations (4) and (5) the discriminant rule which minimises E_∞ has minimum variance from the discriminant vector $\boldsymbol{\rho}$.

Interpreting the prototype target matrix, $\boldsymbol{S}$, as a set of cost vectors,

S_{ji} = *cost of assigning to class j a pattern which belongs to class i*

then $\boldsymbol{\rho}(\boldsymbol{x})$ is the conditional risk vector [4], with i–th component the conditional risk of deciding in favour of class i. The Bayes decision rule for minimum conditional risk is

assign pattern $\boldsymbol{x}$ to class i if $\rho_i(\boldsymbol{x}) \leq \rho_j(\boldsymbol{x})$, $\qquad j = 1, \ldots, c.$

Thus, from Equations (4) and (5), the discriminant rule which minimises E has minimum variance from the optimum Bayes discriminant function, $\boldsymbol{\rho}$, as the number of training samples approaches infinity.

For a 1-from-c target coding scheme, we take

$$S_{ij} = \begin{cases} 1 & \text{if i = j} \\ 0 & \text{otherwise.} \end{cases}$$

This may be viewed as a *gain* matrix in which the gain of assigning to class j a pattern which belongs to class i is zero ($i \neq j$), but is unity for correct classification. The vector $\boldsymbol{\rho}(\boldsymbol{x})$ given by Equation (7) is equal to $\boldsymbol{p}$, the vector of posterior probabilities and the Bayes discriminant rule for minimum error is [6]

$$\textit{assign pattern } \boldsymbol{x} \textit{ to class } i \textit{ if } \rho_i(\boldsymbol{x}) \geq \rho_j(\boldsymbol{x}), \qquad j = 1, \ldots, c.$$

Thus, for this coding scheme, the least-mean-square solution for Λ and $\boldsymbol{\lambda}_0$ gives a vector discriminant function which has minimum variance from the vector of *a posteriori* probabilities. This may also be achieved by taking the target matrix to be the 'equal-cost' matrix,

$$S_{ij} = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{otherwise} \end{cases}$$

or changing the sign of the components S_{ij} to make $\boldsymbol{S}$ a matrix of costs (*i.e.* a cost of -1 for making a correct decision and zero otherwise), and the Bayes decision rule for minimum risk [4, 6].

> Therefore, the least-mean-square approach is optimal for the two particular choices of target matrix in that the discriminant function obtained has minimum variance from the optimal Bayes discriminant function in the limit of an infinite amount of data.

We should make a note on the decision rule here, since the output of a trained pattern recognition system is not $\boldsymbol{\rho}(\boldsymbol{x})$, but the least-mean-square approximation to it given by $\Lambda \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\lambda}_0$ for a pattern $\boldsymbol{x}$. We would like a decision rule which is consistent with the mean-square approach, yet reduces to the Bayes decision rules for minimum error and minimum risk when the targets are chosen appropriately, either as a 1-from-c target coding scheme or a risk vector coding, since in these situations, the discriminant rule is an optimal approximation (in the least-mean-square sense) to the Bayes discriminant vector, $\boldsymbol{\rho}(\boldsymbol{x})$. Since the transformation given by the weights and biases, $\{\boldsymbol{\lambda}\}$, maps the vectors given by the columns of $\boldsymbol{G}$ to the targets, $\boldsymbol{T}$, in a least-squares sense, the logical decision rule (whatever the interpretation of the matrix $\boldsymbol{T}$) is the nearest neighbour decision : decide $\boldsymbol{x}$ belongs to class i if

$$||\boldsymbol{o} - \boldsymbol{s}_i|| \leq ||\boldsymbol{o} - \boldsymbol{s}_j|| \qquad j = 1, \ldots, c \tag{8}$$

where $\boldsymbol{o} = \Lambda \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\lambda}_0$ is the output vector of the system, $\boldsymbol{s}_j$ are the prototype target vectors of class i which form the columns of $\boldsymbol{T}$ and $||\boldsymbol{x}||$ is the Euclidean norm. If there are the same number of patterns as there are classes ($c = P$) then this is the nearest neighbour decision rule.

In the situation where the prototype target vectors are given by a 1-from-c coding, so that the discriminant rule is an optimal approximation to the posterior probabilities, then the decision rule above is equivalent to assigning the vector $\boldsymbol{x}$ to class i if o_i is the largest element of the output vector. Thus the minimum distance decision rule for the approximation $\boldsymbol{o}$ to the optimal vector is identical to the decision rule based on the optimal outputs (i.e. to take the largest component of the discriminant vector as the class).

However, for an arbitrary *loss* matrix as the target matrix, so that the discriminant rule is an optimal approximation to the Bayes conditional risk vector, the minimum distance classifier does not reduce to the Bayes decision for minimum risk (which is to take the smallest component of the discriminant vector as the correct class). Therefore, the most obvious decision rule for mean-square training does not reduce, under limiting cases of the target matrix, to both the Bayes decision rule for minimum risk *and* the Bayes decision rule for minimum error. This is important since it shows that the most commonly used decision rule in traditional statistical pattern analysis, Bayes minimum risk, is *not* in general a special case of the most natural decision rule for this formulation, namely the minimum distance rule.

Returning to Equation (4) for the error, we see that the error E_∞ may be expressed in terms of the error, E', since

$$\begin{aligned} E_\infty &= \sum_{i=1}^{c} p_i \langle \| \boldsymbol{\Lambda G} + \boldsymbol{\lambda}_0 - \boldsymbol{s}_i \|^2 \rangle_i \\ &= \sum_{i=1}^{c} p_i \langle \| \boldsymbol{\Lambda G} + \boldsymbol{\lambda}_0 - \boldsymbol{\rho}(\boldsymbol{x}) + \boldsymbol{\rho}(\boldsymbol{x}) - \boldsymbol{s}_i \|^2 \rangle_i \end{aligned} \tag{9}$$

which may be expanded to give [4]

$$E_\infty = E' + \sum_{i=1}^{c} p_i \langle \| \boldsymbol{s}_i \|^2 \rangle_i - \sum_{i=1}^{c} p_i \langle \| \boldsymbol{\rho}(\boldsymbol{x}) \|^2 \rangle_i + \sum_{i=1}^{c} p_i \langle (\boldsymbol{\Lambda}^* \boldsymbol{G} + \boldsymbol{\lambda}_0)^* (\boldsymbol{\rho}(\boldsymbol{x}) - \boldsymbol{s}_i) \rangle_i \tag{10}$$

The final term in the above expression is identically equal to zero by definition of $\boldsymbol{\rho}$. Therefore we have

$$E_\infty = E' + \sum_{i=1}^{c} p_i \langle \| \boldsymbol{s}_i \|^2 \rangle_i - \sum_{i=1}^{c} p_i \langle \| \boldsymbol{\rho}(\boldsymbol{x}) \|^2 \rangle_i , \tag{11}$$

where, for the target matrix taken to be a matrix of *losses*, the quantity $\sum_{i=1}^{c} p_i \langle \| \boldsymbol{\rho}(\boldsymbol{x}) \|^2 \rangle_i$ is defined as the generalised Bayesian distance [3].

It is straightforward to show that for an arbitrary target matrix

$$\sum_{i=1}^{c} p_i \langle \| \boldsymbol{s}_i \|^2 \rangle_i - \sum_{i=1}^{c} p_i \langle \| \boldsymbol{\rho}(\boldsymbol{x}) \|^2 \rangle_i \geq 0 \tag{12}$$

and so we have

$$E_\infty \geq E' \geq 0 \tag{13}$$

Therefore, the solution obtained by minimising E_∞ gives an error which is an upper bound for E'.

In principle, the error E' may be made arbitrarily small by a suitable choice of the functions $g_i(\boldsymbol{x})$, provided that the rank of $\boldsymbol{\Lambda g}(\boldsymbol{x})$ is not less than the rank of $\boldsymbol{\rho}$. In the following section, we shall consider a particular parametric form for these nonlinear functions, whose parameters may be determined readily using an optimisation strategy. Then, not only may we choose the values of the weights Λ and λ_0 to ensure that E_∞ is a minimum in the space spanned by these weights, but also we may choose the parameters of the functions g_i to ensure that the error is a minimum in the space spanned by the weights *and* the parameters of g_i.

However, first it must be emphasised that in practice we do not minimise the error, E_∞, given by Equation (4), but rather the finite sample approximation to it given by Equation (3) and the solution for the weights may be written down as follows. Minimising E with respect to the bias vector, λ_0, gives

$$\lambda_0 = \bar{\boldsymbol{t}} - \Lambda\bar{\boldsymbol{g}} \tag{14}$$

where $\bar{\boldsymbol{t}}$ is the mean target vector given by

$$\bar{\boldsymbol{t}} = \frac{1}{P}\sum_{i=1}^{c} n_i \boldsymbol{s}_i \tag{15}$$

and $\bar{\boldsymbol{g}}$ is the mean vector of nonlinearity outputs

$$\bar{\boldsymbol{g}} = \frac{1}{P}\sum_{p=1}^{P} \boldsymbol{g}(\boldsymbol{x}^p). \tag{16}$$

Using this solution for λ_0, the error may be written

$$E = \frac{1}{P}\|\hat{\boldsymbol{T}} - \Lambda\hat{\boldsymbol{G}}\|^2, \tag{17}$$

where $\hat{\boldsymbol{T}}$ and $\hat{\boldsymbol{G}}$ are defined as

$$\begin{aligned} \hat{\boldsymbol{T}} &\triangleq \boldsymbol{T} - \bar{\boldsymbol{t}}\mathbf{1}' \\ \hat{\boldsymbol{G}} &\triangleq \boldsymbol{G} - \bar{\boldsymbol{g}}\mathbf{1}' \end{aligned} \tag{18}$$

where $\mathbf{1}$ is a $P \times 1$ vector of 1's.

The solution for Λ which minimises E with minimum (Frobenius) norm is

$$\Lambda = \hat{\boldsymbol{T}}(\hat{\boldsymbol{G}})^+, \tag{19}$$

where $\hat{\boldsymbol{G}}^+$ is the Moore-Penrose pseudo-inverse of $\hat{\boldsymbol{G}}$ [7].

In the following section we exploit this solution and the properties of the pseudo-inverse to show that minimising the error is equivalent to maximising a particular feature extraction criterion in the space spanned by the parameters of the functions g_i [21, 3].

3 Networks and Nonlinear Discriminant Analysis

In this section we shall consider a particular form for the nonlinearities in the discriminant function, namely a layered feed-forward classification network. This class of structures attempts to map a set of P, n-dimensional feature patterns onto a set of P, n'-dimensional target patterns. The set of P feature-target pattern pairs constitutes the training set. In its simplest form, we consider a network with n input nodes which are fully connected to a hidden layer of n_0 nodes with arbitrary nonlinear transfer functions. The input to each node may be taken to be the scalar product between the pattern produced at the outputs of the previous layer and the set of weights attached to the links of that node. However, other 'combination rules' may also be used [2]. These hidden nodes are fully connected to a set of n' output nodes and for purposes of classification, we consider that the number of output nodes $n' = c$, the number of classes. We could consider arbitrary numbers of layers, nonlinearities and combination rules (between the weights in one layer and the output patterns of the previous layer) without affecting the analysis. However, we restrict the network to one with output units which have linear transfer functions, and so the transformation from the patterns created in the space of the hidden units to the output space is linear. Thus, the output of the k-th output node for input pattern $\boldsymbol{x}$ is identical to the generalised linear discriminant function Equation (1),

$$O_k = \lambda_{0k} + \sum_{j=1}^{n_0} \lambda_{jk} g_j(\boldsymbol{x}) \tag{20}$$

where λ_{jk} are now viewed as the connections from the j-th hidden unit to the k-th output unit (λ_{0k} is a bias term for the k-th output) and g_j is the nonlinear function of the input associated with the j-th hidden node. For a single hidden-layer multilayer perceptron with nonlinear transfer functions, ϕ_j associated with the j-th hidden unit (often taken to be a logistic function), and inputs to each node given by a scalar product of the input with a vector of weights attached to the links of that node, Equation (20) may be written as

$$O_k = \lambda_{0k} + \sum_{j=1}^{n_0} \lambda_{jk}\phi_j\left(\mu_{0j} + \sum_{i=1}^{n} x_i\mu_{ij}\right), \tag{21}$$

where x_i is the i-th component of the input feature vector to the network, $\boldsymbol{x}$, and μ_{ij} are the weights in the first layer connecting the i-th input to the j-th hidden node.

Training consists of adapting the parameters of the network (the weights and biases) by any suitable optimisation strategy [20] (since the function and it's derivative as determined by the network are known analytically from (20)) to minimise the square error between the desired target patterns and the actual output patterns of the network over the entire training set. *Generalisation* ability depends on the network being complex enough (as determined by the number of hidden units) to model adequately the structure in the training data without being over-complex which would allow the network to fit the superimposed noise on the data.

Training a network usually involves the minimisation of a sum-square error of the form

$$E = \frac{1}{P}\| \boldsymbol{T} - \boldsymbol{O} \|^2 \tag{22}$$

where $\boldsymbol{T}$ is the $c \times P$ matrix of target vectors and $\boldsymbol{O}$ is the $c \times P$ matrix of network output vectors. We have seen in Section 2 that this criterion is optimum in the sense that the solution obtained for the final layer weights (in the limit of $P \rightarrow \infty$) yields a discriminant function which (for a set of loss vectors as target vectors) has minimum variance from the optimum Bayes solution for minimum risk. However, in the feed-forward network considered here we are not simply optimising the final layer weights, but also the parameters of the nonlinearities (the feature extraction functions) by adjusting the first layer weights. All of these weights may be adjusted simultaneously by some suitable optimisation scheme. The one most commonly used is steepest descents, but others have been considered and their performance on a range of data sets assessed [19]. An alternative approach is to consider the error to be a function of the first layer weights alone, choosing the final layer weights, $\{\lambda\}$, using Equations (14) and (19). The optimisation strategy then adjusts the first layer weights and for each adjustment solves Equations (14) and (19) for the final layer weights. This ensures that the error is always a minimum with respect to the final layer weights. Thus, the optimisation strategy is to choose the parameters of the nonlinearities, $\{\mu\}$, using some nonlinear optimisation strategy so that the optimum linear discriminant error, achieved by the final linear transformation, is minimised. This hybrid linear–nonlinear optimisation strategy has been assessed in [20].

The advantage of representing the nonlinear functions, g_j, as a transformation performed by a feed-forward layered network is that we may easily exploit the functional form in order to minimise the output error. Of course, this amounts to nothing more than the chain rule for differentiation and a gradient rule for function minimisation and it is in 'operation', when a network is required to perform real-time classification or control, that network architectures will give most gain. However, parameterisation of the nonlinear functions, by whatever means, leads to increased complexity of the pattern recognition algorithm. Although this may give an improved performance on a training set, performance on an unseen test set (generalisation) may be poor due to possible overtraining. In pattern recognition, there has been considerable research devoted to the trade–off between dimensionality, sample size and algorithm complexity [11]. For example, one might naïvely expect that as the dimension of a measurement vector is increased, the classification performance should also increase. However, in practice, the performance of a classifier is seen to improve up to a point and then decrease. This is also expected to apply to network-based classifiers for which the number of adjustable parameters increases as the input dimension increases. Therefore, as with any problem in estimation, a natural requirement is that the number of samples available significantly exceeds the number of model parameters. For a feed-forward network, a relationship between the number of training nodes, n_0, the number of training patterns, P, and the maximum number of separable regions has been given in [14], but rules relating the dimension, sample set size and complexity to classifier performance are, so far, elusive.

The error equation (22) may be written in a more general form if we consider the optimum effects of minimising a *weighted* error function where the p-th training pattern may be weighted by the real factor d_p,

$$\begin{aligned} E &= \frac{1}{P}\sum_{p=1}^{P} d_p \| \boldsymbol{t}^p - \boldsymbol{o}^p \|^2, \\ &= \frac{1}{P}\|(\boldsymbol{T} - \boldsymbol{O})\boldsymbol{D}\|^2 \\ &= \frac{1}{P}\|(\boldsymbol{T} - \boldsymbol{\Lambda}\boldsymbol{H})\boldsymbol{D}\|^2 \end{aligned} \tag{23}$$

where $\boldsymbol{o}^p$ is the p-th output vector and p-th column of the $c \times P$ matrix $\boldsymbol{O}$; the matrix $\boldsymbol{D}$ is a $P \times P$ diagonal matrix whose elements are $\sqrt{d_p}$; and $\boldsymbol{H}$ is the $n_0 \times P$ matrix of hidden unit outputs.

In the following section we shall examine various combinations of target coding methodology and suitable choices for the error weighting factors d_p, which allow the performance of a network in operation to be tailored by prior knowledge.

Because the output nodes have linear transfer functions, the linear transformation performed by the final layer may be inverted by pseudo–inverse methods to reveal the optimum distribution of patterns in the n_0–dimensional space at the output of the hidden units which enables the minimum error to be achieved. It is found (see Appendix A for details) that the choice of weights in the initial layer projects the training patterns by a nonlinear transformation into a distribution at the output of the hidden layer so as to maximise a feature extraction criterion given by

$$J = Tr\left\{S_B S_T^+\right\} \tag{24}$$

where Tr denotes the trace of a matrix, S_T^+ is the pseudo–inverse of S_T and the matrices S_B, S_T are defined as

$$S_T \triangleq \frac{1}{P}\hat{H}D^2\hat{H}^* \tag{25}$$

and

$$S_B \triangleq \frac{1}{P^2}\hat{H}D^2\hat{T}^*\hat{T}D^2\hat{H}^* \tag{26}$$

where $\hat{H}$, $\hat{T}$ are defined in Appendix A to be the *mean–shifted* set of patterns at the output of the hidden layer and the *mean–shifted* set of target patterns on the training set (where the mean is weighted by the prior expectations). The matrices S_T and S_B have the interpretation of being the total and between class covariance matrices *of the output patterns of the hidden units.* Precise interpretations depend on specific target coding schemes and weighting factors which will be considered in the following section. Thus the optimum method of solution of such a network is to find a nonlinear transformation into the space spanned by the hidden units such that the patterns in different classes are somehow maximally separated (this information is contained in the between class covariance matrix), while still maintaining an overall total normalisation (through the total covariance matrix). In this sense, feed–forward layered networks operating as classifiers succeed because they perform a specific nonlinear discriminant analysis by exploiting subspace methods.

The criterion (24) is independent of the transformation from the data space to the space of hidden unit outputs; i.e. it is not dependent on the layered structure described above, but it is a property of the least–mean–square solution for the final layer. A similar expression has been described by Devijver [3] who points out that one of its main advantages is that it is a feature extraction criterion which takes into account the costs of misclassification. The expression is also related to optimisation criteria used in clustering [10]. What we have shown here is that the first part (up to the outputs of the final layer of hidden units) of a feed–forward network with linear transfer functions at the outputs performs feature extraction which maximises the criterion (24), and the subsequent final layer performs an optimum mapping on to the targets. Thus, optimising a feed–forward network using a least–mean–square approach optimises a specific feature extraction and performs classification simultaneously, whereas conventionally these operations have been addressed separately. Performing feature extraction matched to discrimination is one reason why adaptive networks have been demonstrated to produce good classification performance over a range of problems.

An advantage of solving for the final layer weights using a pseudo–inverse approach (such that the final layer weights have minimum norm) is that the output values for *any* subsequent input pattern satisfy a particular linear constraint which depends on the target coding (see Appendix B). In particular, for a 1–from–c target coding scheme, the outputs sum to unity for *any* given input pattern. For this target coding scheme, not only does the final transformation approximate the posterior probabilities in a least-squares sense, but also gives outputs with the additional property that they are guaranteed to sum to unity. However, it may not be possible, necessarily, to interpret the outputs individually as probabilities, since the additional requirement for classical probabilities that the outputs lie between zero and unity may not be satisfied.

In this section we have described one scheme (a feed-forward layered network) for parameterising the nonlinearities of a generalised discriminant function. This allows optimisation methods to be used in order for the subsequent linear transformation to be a better approximation (in the case of 1-from-c coding for example) to the posterior probabilities. In addition, we have shown that minimising the error maximises a particular feature extraction criterion at the outputs of the hidden units, and the pseudo-inverse approach naturally leads to outputs which satisfy a linear constraint. More importantly, for 1-from-c coding, the outputs sum to unity.

In the next section, we shall consider various combinations of target coding and error weighting. For each combination, we shall evaluate and interpret the feature extraction cost function, J; we shall derive the 'optimal' solution in the limit of infinite samples, ρ; we shall evaluate the constraint satisfied by the outputs and derive an appropriate decision rule.

4 Particular Pattern Weighting and Coding Schemes

4.1 Example 1

Choose uniform weighting for each pattern in the training set ($d_p = 1, p = 1, \ldots, P$). In this case, the matrix S_T may be expanded out explicitly as

$$S_T \equiv \frac{1}{P}\sum_{p=1}^{P}\left(\boldsymbol{\phi}^p - \boldsymbol{m}^H\right)\left(\boldsymbol{\phi}^p - \boldsymbol{m}^H\right)^* \tag{27}$$

where $\boldsymbol{\phi}^p$ is the hidden layer output pattern vector for pattern p, and $\boldsymbol{m}^H$ is the *mean* pattern vector (defined over the training set) at the output of the hidden units whose j-th component may be expressed as $m_j^H = \sum_{p=1}^{P} \phi_j^p / P$. Equation (27) is the traditional total covariance matrix of the hidden unit output patterns evaluated over the entire training set.

For a 1-from-c target coding scheme, the matrix S_B may be expanded explicitly as

$$S_B \equiv \sum_{k=1}^{c}\left(\frac{n_k}{P}\right)^2\left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)\left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)^* \tag{28}$$

where n_k is the number of patterns in class k, and $\boldsymbol{m}_k^H$ is the mean pattern vector over all hidden unit outputs which belong to class k in the training set,

$$\boldsymbol{m}_k^H = \frac{1}{n_k}\sum_{\boldsymbol{\phi}^p \in k} \boldsymbol{\phi}^p$$

Equation (28) is recognised as the *weighted* (weighted by the square of the numbers of patterns in each class) between class covariance matrix. Thus, for this particular target coding scheme, the classes with largest membership dominate the transformation.

The optimal discriminant vector (in the sense that choosing the largest component as indicative of the correct class gives a maximum *gain* decision), in the limit of infinite samples, for a given input pattern, $\boldsymbol{x}$, is given by Equation (6) and for the 1-from-c coding it has components

$$\rho_i \;=\; p(i|\boldsymbol{x}), \tag{29}$$

the posterior probabilities.

Also, for this target coding scheme, the linear constraint satisfied by the output, $\boldsymbol{o} = (o_1, o_2, \ldots, o_c)^*$, of the network, for an arbitrary input pattern, is

$$\sum_{i=1}^{c} o_i \;=\; 1. \tag{30}$$

This is a property required by a network which encodes probabilities as outputs, though the additional property $0 \leq o_i \leq 1, i = 1, \ldots, c$ cannot be guaranteed.

The minimum distance decision rule may be written as

$$\textit{assign } \boldsymbol{x} \textit{ to class } i \textit{ if } o_i \geq o_j, \qquad j = 1, \ldots, c,$$

in other words, select the largest output as the correct class. This is also the decision rule which would apply if the optimal outputs $p(i|\boldsymbol{x})$ were attained by the network.

4.2 Example 2

Choose uniform weighting for each pattern in the training set and employ a target coding scheme which is reciprocally weighted by the numbers in the class. Specifically, consider the target value of class k for pattern p

$$t_k^p = \begin{cases} (P/n_k)^{\frac{1}{2}} & \text{if } \boldsymbol{x}^p \in k \\ 0 & \text{otherwise} \end{cases}$$

This form of output coding means that the *gain* in achieving a correct classification is inversely proportional to the square root of the proportions of samples in each class in the training set. The matrix $\boldsymbol{S_T}$ is again the covariance matrix of hidden unit outputs, but in this case, the matrix $\boldsymbol{S_B}$ expands to

$$\boldsymbol{S_B} = \sum_{k=1}^{c} \frac{n_k}{P} \left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)\left(\boldsymbol{m}_k^H - \boldsymbol{m}^H\right)^* \tag{31}$$

which is the *conventional* between class covariance matrix. Thus, employing the above target coding scheme produces a feature extraction criterion which prevents a network from over-compensating for uneven class distributions.

The components of the optimal discriminant vector (again, in the sense that choosing the largest component as indicative of the correct class gives a maximum *gain* decision), are given by

$$\rho_i \;=\; \sqrt{\frac{P}{n_i}}\, p(i|\boldsymbol{x}) \tag{32}$$

and the constraint satisfied by the outputs of the network is

$$\sum_{i=1}^{c} \sqrt{\frac{n_i}{P}} o_i = 1. \tag{33}$$

The minimum distance decision rule is to assign x to class i if

$$\frac{P}{n_i}\left(1 - 2\sqrt{\frac{n_i}{P}} o_i\right) \leq \frac{P}{n_j}\left(1 - 2\sqrt{\frac{n_j}{P}} o_j\right), \qquad j = 1, \ldots, c,$$

If the network were to achieve the optimal output, so that $o_i = \rho_i$, then by Equation (32) $\sqrt{n_i/P} o_i$ would be equal to the posterior probabilities and the above decision rule is to assign x to class i if

$$\frac{P}{n_i}(1 - 2p(i|x)) \leq \frac{P}{n_j}(1 - 2p(j|x)) \qquad j = 1, \ldots, c.$$

This is one of the important differences between a 1–from–c coding scheme and a $\sqrt{(\frac{P}{n_k})}$–from–c coding scheme. In the former case, for a network achieving the optimal Bayes discriminant vector, the optimal 'maximum gain' decision is the same as the minimum distance decision. In the latter situation, if the network achieves an output vector which is equal to the Bayes vector for maximum gain, then the minimum distance decision does not give a Bayes maximum gain.

4.3 Example 3

Choose a uniform weighting for each pattern and employ a target coding scheme which, for a pattern in class i, takes as the target vector the vector s_i, whose j–th component is the cost of assigning to class j a pattern which belongs to class i. In this case, the matrix S_B is given by

$$\sum_{j=1}^{c}\left(\sum_{k=1}^{c} \frac{n_k}{P} S_{jk}(m_k^H - m^H)\right)\left(\sum_{k=1}^{c} \frac{n_k}{P} S_{jk}(m_k^H - m^H)\right)^*, \tag{34}$$

where S is the prototype target matrix with i–th column s_i.

The optimal discriminant vector is

$$\rho(x) = \sum_{i=1}^{c} s_i p(i|x),$$

which is the Bayes conditional risk vector.

The minimum distance decision rule is to assign x to class i if

$$s_i^* s_i - 2o^* s_i \leq s_j^* s_j - 2o^* s_j \qquad j = 1, \ldots, c$$

Generally, this is not the same as the Bayes minimum risk decision rule.

4.4 Example 4

Weight each pattern in the training set according to the *a priori* class probabilities and the number in that class according to

$$d_p = \frac{P_k}{n_k/P} \qquad \textit{for pattern p in class k} \tag{35}$$

where P_k is the class probability (derived from prior knowledge regarding the relative expected class importance, or frequency of occurrence in operation) and n_k is the number of training patterns in class k in the training set. Using this error weighting the total covariance matrix may be expressed as

$$S_T = \sum_{k=1}^{n'} \frac{P_k}{n_k} \sum_{\phi^p \in k} \left(\phi^p - m^H\right)\left(\phi^p - m^H\right)^* \tag{36}$$

This is the sample-based estimate of the mixture covariance matrix. In this expression, the vector m^H is the sample based estimate of the population mean which may be expressed as

$$\begin{aligned} m^H &= \sum_{k=1}^{n'} \frac{P_k}{n_k} \sum_{\phi^p \in k} \phi^p \\ &= \sum_{k=1}^{n'} P_k m_k^H \end{aligned} \tag{37}$$

where

$$m_k^H \triangleq \frac{1}{n_k} \sum_{\phi^p \in k} \phi^p \tag{38}$$

is the estimate of the mean of class k.

If a 1-from-c target coding scheme is employed (as in Example 1 above) along with the above pattern weighting scheme, the between class covariance matrix is modified to

$$S_B \equiv \sum_{k=1}^{n'} P_k^2 \left(m_k^H - m^H\right)\left(m_k^H - m^H\right)^* \tag{39}$$

where m_k^H and m^H are given in (37) and (38). This may be interpreted as a sample-based estimate of the weighted between class covariance matrix. Similarly, choosing to weight the targets according to a $1/\sqrt{P_k} - \mathit{from} - n'$ coding along with the above pattern weighting, leads to a sample based estimate of the *conventional* between class covariance matrix (Equation (31) above, but with n_k/P replaced by the prior probabilities P_k and taking m_k^H and m^H as defined by Equations (37) and (38)).

Note that the limit of the prior probabilities being proportional to the numbers in each class reproduces the 'standard' model. This analysis indicates how a combination of target weighting and error weighting by the inclusion of prior probabilities in the training scheme, induces a nonlinear transformation which is capable of compensating for different class importance or pattern distributions between classes. For instance, if in a c-class problem the occurrence of each class in operation is considered equally likely but the number of patterns in each class in the training set is distributed with n_k in class k, then one should weight each training pattern according to

$$d_p = \frac{P}{c \times n_k} \qquad \mathit{for\ pattern\ p\ in\ class\ k} \tag{40}$$

In addition, using a $1/\sqrt{P_k} - \mathit{from} - c$ target coding scheme ensures that the nonlinear transformation into the space of the hidden units involves the conventional between class covariance matrix.

The optimal discriminant vector (in the limit of infinite samples from the training set, sampled at proportions different from the test set) has components

$$\rho_i \; = \; \sum_{i=1}^{c} s_i p'(i|x) \tag{41}$$

where $p'(i|x)$ is the posterior probability in the test set given a pattern x.

Thus, for a 1-from-c training scheme, ρ_i is the probability of class i given the test data. For a general input, the outputs still sum to unity for this error weighting and the minimum distance decision rule corresponds to the Bayes decision rule for minimum expected error.

5 Discussion.

In this paper we have viewed the problem of classification with feed-forward networks as an optimisation problem using generalised linear discriminant functions. The weights of the discriminant functions and the parameters of the nonlinear functions may be adapted simultaneously using a hybrid linear-nonlinear optimisation strategy to give a minimum output error. Thus, the network performs adaptive feature extraction matched to the optimal least-mean-squares classification. We have shown that minimising the sum-square error at the output of the network is equivalent to maximising a specific feature extraction criterion, J, at the outputs of the hidden units given by

$$J \; = \; Tr\{S_B S_T^+\}$$

where S_B and S_T may be interpreted as the between-class and total covariance matrices of hidden unit outputs. The precise interpretation depends on the error weighting and target coding scheme. We have also exploited relationships between deterministic networks and statistical decision theory to show that the minimum error in the infinite sample limit is an upper bound for the error for the Bayes discriminant function. Therefore training a multilayer perceptron maximises a feature extraction criterion and minimises an upper bound for the Bayes error.

The choice of error weighting and target coding is governed by prior knowledge of the classification task. We highlighted two important points about real-world data and how to compensate for their effects in network training. The first point is that in many pattern processing tasks the availability of data constituting a training set may not reflect the expected distribution of patterns 'in operation'. In order to maximise the likelihood of performing correct classification *on the test set* the training set patterns need to be weighted by class-conditional probabilities. The effect that this weighting has on the feature extraction criterion employed in the network was discussed in Section 4.4. The second point is that, even if the distribution of patterns in the training set can be considered 'representative', there is usually an anisotropic penalty associated with misclassifying patterns. Thus, if it is a more serious error to misclassify a pattern in class (i) into class (j), rather than a pattern from class (j) into class (i), then the associated 'costs' of these misclassifications may be incorporated into a prototype target matrix. The effect of folding costs into the training phase itself was considered in Sections 4.2 and 4.3 and related to the minimum Bayes risk decision in Section 2.

In addition, it has been shown that for an optimisation scheme which solves for the final layer weights using a pseudo-inverse method, the outputs of the network satisfy certain constraints. In particular, for a 1-from-c target coding scheme, the pseudo-inverse approach for the final-layer weights ensures that the outputs sum to unity. This hybrid linear-nonlinear optimisation method has been described elsewhere [20].

The main advantages of the least-squares approach is that it assumes no prior knowledge of the probability distribution of the patterns. However, it places emphasis on regions of highest sample density, rather than on regions near the decision boundary. Several approximations have been proposed in the literature which remedy this (see, for example, [4]). For example, the error weighting introduced in Section 3 may be a function which reaches a maximum at the boundaries of the decision region. However, in the types of network considered in this paper, the use of logistic transfer functions for the hidden nodes means that the boundary is mainly influenced by data points near to it. A further problem is that in the classical least-square problem, there is an underlying assumption that the errors are in the target matrix, $\boldsymbol{T}$, whereas in reality there will be errors in the training set due to noise on the data and measurement errors. In such a situation, a total least-squares approach should be considered [8].

In conclusion, this paper has elucidated why networks perform well as static pattern classification devices and pointed out situations where their performance is biased by the prior distribution of patterns in the training set. A general regularisation scheme which compensates for the discussed uneveness has been proposed, and the consequences that the coding scheme has on the feature extraction criterion have been presented analytically. Detailed numerical simulations confirm the generic statements made in this paper and are presented elsewhere [12].

Appendix A The Generalised Network Feature Extraction Criterion.

Minimising the weighted error, Equation (23), with respect to the bias on the outputs, λ_0, gives the solution

$$\lambda_0 = \bar{t} - \Lambda m^H, \tag{42}$$

where $\bar{t}$ and m^H are given by

$$\begin{aligned} \bar{t} &= \frac{TD^2 1}{\sum_{p=1}^{P} d_p} \\ m^H &= \frac{HD^2 1}{\sum_{p=1}^{P} d_p} \end{aligned} \tag{43}$$

and H is the $n_0 \times P$ matrix of hidden unit outputs. Substituting for λ_0 into the equation for the error gives

$$E = \frac{1}{P} \|(\hat{T} - \Lambda \hat{H})D\|^2, \tag{44}$$

with the definitions

$$\hat{T} \triangleq T - \bar{t}1^* \tag{45}$$

and

$$\hat{H} \triangleq H - m^H 1^* \tag{46}$$

The matrix Λ which minimises the error, E, with minimum norm is given by

$$\Lambda = \hat{T}D(\hat{H}D)^+, \tag{47}$$

where $(\hat{H}D)^+$ denotes the Moore–Penrose pseudo–inverse of the matrix $\hat{H}D$. Substituting for Λ into Equation (44) and using the properties of the pseudo–inverse, the error may be written [20]

$$E = \frac{1}{P} Tr\{\hat{T}D^2\hat{T}^* - \hat{T}D^2\hat{H}^*(\hat{H}D^2\hat{H}^*)^+\hat{H}D^2\hat{T}^*\}, \tag{48}$$

where Tr is the matrix trace operation.

Thus, since the targets, T, and the weights, D, are fixed, minimising the error, E, is equivalent to maximising the discriminant function

$$J = \frac{1}{P} Tr\{\hat{T}D^2\hat{H}^*(\hat{H}D^2\hat{H}^*)^+\hat{H}D^2\hat{T}^*\}, \tag{49}$$

at the outputs of the hidden units. Equation (49) may be rearranged to give [20]

$$J = Tr\{S_B S_T^+\}, \tag{50}$$

where

$$S_T \triangleq \frac{1}{P}\hat{H}D^2\hat{H}^*, \tag{51}$$

and

$$S_B \triangleq \frac{1}{P^2}\hat{H}D^2\hat{T}^*\hat{T}D^2\hat{H}^*. \tag{52}$$

This generalises the result in [20] where the matrix D was taken to be the identity matrix. The result states that minimising a weighted sum–squared error at the outputs of a feed-forward network with linear output units, and in which the final layer weights are chosen by a minimum norm solution, maximises a feature extraction criterion (given by Equation (50)) at the outputs of the hidden units. The feature extraction criterion is the trace of a product of a matrix S_B with the pseudo-inverse of a matrix S_T.

Appendix B Sum Rules

Theorem.

Consider a network having linear output units. Let the weights associated with the connections to these units be determined by linear minimum norm least squares optimisation. Then if there exists an arbitrary linear constraint of the form

$$\boldsymbol{u}^{*}\boldsymbol{t}^{p} = \boldsymbol{u}^{*}\bar{\boldsymbol{t}} \qquad \forall\, p = 1, 2, \ldots, P$$

with $\boldsymbol{u}$ *a constant vector, then the general output* $\boldsymbol{o}$ *of the network satisfies:*

$$\boldsymbol{u}^{*}\boldsymbol{o} = \boldsymbol{u}^{*}\bar{\boldsymbol{t}}$$

Proof

The general output of the network is given by

$$\boldsymbol{o} = \lambda_0 + \Lambda \boldsymbol{h}$$

where $\boldsymbol{h}$ is the output of the hidden units. Using the solution for λ_0, Equation (42), and for Λ, Equation (47), this may be written

$$\boldsymbol{o} = \bar{\boldsymbol{t}} + \hat{\boldsymbol{T}}\boldsymbol{D}(\hat{\boldsymbol{H}}\boldsymbol{D})^{+}\left(\boldsymbol{h} - \boldsymbol{m}^{H}\right)$$

Therefore

$$\boldsymbol{u}^{*}\boldsymbol{o} = \boldsymbol{u}^{*}\bar{\boldsymbol{t}} + \boldsymbol{u}^{*}\hat{\boldsymbol{T}}\boldsymbol{D}(\hat{\boldsymbol{H}}\boldsymbol{D})^{+}\left(\boldsymbol{h} - \boldsymbol{m}^{H}\right)$$

But

$$\boldsymbol{u}^{*}\hat{\boldsymbol{T}} = \boldsymbol{u}^{*}\boldsymbol{T} - \boldsymbol{u}^{*}\bar{\boldsymbol{t}}\boldsymbol{1}^{*}$$

By hypothesis, $\boldsymbol{u}^{*}\boldsymbol{T} = \boldsymbol{u}^{*}\bar{\boldsymbol{t}}\boldsymbol{1}^{*}$. Therefore

$$\boldsymbol{u}^{*}\boldsymbol{o} = \boldsymbol{u}^{*}\bar{\boldsymbol{t}} \quad \blacksquare$$

Remark: If the set of target vectors satisfy several linear constraints simultaneously, then so will the general network outputs.

As a special case, consider the quantity $\boldsymbol{1}^{*}\hat{\boldsymbol{T}}$ in the situation when the sum of each column of $\boldsymbol{T}$ is a constant ($= t$ say);

$$\begin{aligned}
\boldsymbol{1}^{*}\hat{\boldsymbol{T}} &= \boldsymbol{1}^{*}\boldsymbol{T} - \boldsymbol{1}^{*}\bar{\boldsymbol{t}}\boldsymbol{1}^{*} \\
&= \boldsymbol{1}^{*}\boldsymbol{T} - \frac{\boldsymbol{1}^{*}\boldsymbol{T}\boldsymbol{D}^{2}\boldsymbol{1}\boldsymbol{1}^{*}}{\sum_{\mathrm{p}=1}^{\mathrm{P}} \mathrm{d}_{\mathrm{p}}} \\
&= t\boldsymbol{1}^{*} - t\frac{\boldsymbol{1}^{*}\boldsymbol{D}^{2}\boldsymbol{1}\boldsymbol{1}^{*}}{\sum_{\mathrm{p}=1}^{\mathrm{P}} \mathrm{d}_{\mathrm{p}}} \\
&= \boldsymbol{0}^{*}
\end{aligned} \tag{53}$$

since $\boldsymbol{1}^{*}\boldsymbol{D}\boldsymbol{1} = \sum d_p$.

Therefore, the sum over the outputs of the trained, optimised network is given by

$$\begin{aligned} \text{Sum over outputs} &= \\ \mathbf{1}^{*}\mathbf{o} &= \mathbf{1}^{*}\mathbf{t} \\ &= t, \ \text{Sum of each column} \end{aligned} \tag{54}$$

This completes the proof of the original observation. In particular, for a 1-from-c coding scheme where the components of the target vector belong to $\{1, 0\}$, the sum of the outputs of the network for *any* input sum to unity.

References

[1] D.G. Bounds, B. Mathew, and G. Waddell, : "A Multi-layer Perceptron Network for the Diagnosis of Low Back Pain", *IEEE Int. Conf. on Neural Networks, California, 1988*, **II**, II–481—II–489, 1988.

[2] D.S. Broomhead, and D. Lowe, : "Multi-variable Functional Interpolation and Adaptive Networks", *Complex Systems,* **2**, No. 3, 269-303, 1988.

[3] P.A. Devijver, : "Relationships Between Statistical Risks and the Least-mean-square-error Design Criterion in Pattern Recognition", *Proc. First Int. Joint Conf. Pattern Recognition, Washington, November 1973,* 139-148, 1973.

[4] P.A. Devijver, and J. Kittler, : "Pattern Recognition : A Statistical Approach", Prentice-Hall International, Inc., London, 1982.

[5] R.A. Fisher, : "The Use of Multiple Measurements in Taxonomic Problems", *Annals of Eugenics*, 7,179-188, 1936.

[6] K. Fukunaga, : "Introduction to Statistical Pattern Recognition", Academic Press, New York, 1972.

[7] G. Golub, and W. Kahan, : "Calculating the Singular values and Pseudo-inverse of a Matrix", *SIAM Journal Numerical Analysis,* **2** (2), 205-224. 1965.

[8] G.H. Golub, and C.F. Van Loan, : "An Analysis of the Total Least Squares Problem", *SIAM Journal Numerical Analysis*, **17** (6), 883-893, 1980.

[9] R.P. Gorman, and T.J. Sejnowski, : "Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets", *Neural Networks*, **1**, (1), 75-90, 1989.

[10] D.J. Hand, : "Discrimination and Classification", John Wiley and Sons, 1981.

[11] A.K. Jain, and B. Chandrasekharan, "Dimensionality and Sample size Considerations in Pattern Recognition Practice", *Handbook of Statistics*, **2**, Krishnaiah, P.R. and Kanal, L.N. (eds), North-Holland Publishing Company, 835-855, 1982.

[12] D. Lowe, and A.R. Webb, : "Encoding Prior Probabilities and Misclassification costs into Network Training : An Example From Medical Prognosis", *RSRE Memorandum 4343*, 1989.

[13] W.S. Meisel, : "Least-square Methods in Abstract Pattern Recognition", *Information Sciences*, **1**, 23-42, 1968.

[14] G. Mirchandani, and W. Cao, : "On Hidden Nodes for Neural Networks", *IEEE Trans. Circuits and Systems*, **36**, No. 5, 661-664, 1989.

[15] N.J. Nilsson, : "Learning Machines : Foundations of Trainable Pattern-classifying Systems", McGraw-Hill, Inc, 1965.

[16] S.M. Peeling, R.K. Moore, and M.J. Tomlinson, : "The multi-layer perceptron as a tool for speech pattern processing research", *Proceedings IoA Autumn Conference on Speech and Hearing,* **8**, 307-314, 1986.

[17] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, : "Learning internal representation by error propagation", in *Parallel distributed processing: Explorations in the microstructure of cognition*, (Vols, 1 and 2), Cambridge, MA:MIT Press, 1986.

[18] D.F. Specht, : "Generation of Polynomial Discriminant Functions for Pattern Recognition", *IEEE Trans. Electronic Computers*, **EC-16**, No. 3, 308-319, 1967.

[19] A.R. Webb, and D. Lowe, : "A Comparison of Nonlinear Optimisation Strategies for Feed-forward Adaptive Layered Networks", *RSRE Memorandum 4157*, 1988.

[20] A.R. Webb, and D. Lowe, : "A Hybrid Optimisation Strategy for Adaptive Feed-forward Layered Networks", *RSRE Memorandum 4193*, 1988.

[21] A.R. Webb, and D. Lowe, : "The Optimised Internal Representation of Multilayer Classifier Networks Performs Nonlinear Discriminant Analysis", *Neural Networks*, to appear, 1989

[22] W.G. Wee, : "Generalized Inverse Approach to Adaptive Multiclass Pattern Classification", *IEEE Trans on Computers*, **C-17**, No. 12, 1157-1164, 1968.

[23] S.S. Yau, and J.M. Garnett, : "Least-mean-square Approach to Pattern Classification", in M.S. Wantanabe (ed.) *Frontiers of Pattern Recognition*, New York, Academic Press, 1972.

[24] T.Y. Young, and T.W. Calvert, : "Classification, Estimation and Pattern Recognition", American Elsevier Publishing Company, New York, 1974.

DOCUMENT CONTROL SHEET

Overall security classification of sheet ............UNCLASSIFIED............

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification, eg (R), (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference MEMO 4342	3. Agency Reference	4. Report Security Classification UNCLASSIFIED

5. Originator's Code (if known) 7784000	6. Originator (Corporate Author) Name and Location ROYAL SIGNALS & RADAR ESTABLISHMENT ST ANDREWS ROAD, GREAT MALVERN WORCESTERSHIRE WR14 3PS
5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title

ON NETWORKS, OPTIMISED FEATURE EXTRACTION AND
THE BAYES DECISION

7a. Title in Foreign Language (in the case of Translations)

7b. Presented at (for Conference Papers) Title, Place and Date of Conference

8. Author 1 Surname, Initials LOWE D	9a. Author 2 WEBB A	9b. Authors 3, 4	10. Date 1989.12	pp ref 21

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution Statement

UNLIMITED

Descriptors (or Keywords)

Continue on separate piece of paper

Abstract

In this paper we address the problem of multi-class pattern classification using adaptive layered networks. We view such networks as performing generalised linear discriminant analysis in which a particular parametric form is assumed for the nonlinear functions. Training the network consists of a least-square approach which combines a generalised inverse computation to solve for the final layer weights, together with a nonlinear optimisation scheme to solve for parameters of the nonlinearities. Such an approach performs feature extraction and classification simultaneously, in which the feature extraction is (optimally) matched to the classification scheme. We derive a general analytic form for the feature extraction criterion and interpret it for specific forms of target coding and error weighting. A particular aspect of the approach is to exhibit how a priori information regarding non-uniform class membership, uneven distribution between train and test sets and misclassification costs may be exploited in a regularised manner in the training phase of networks.